

# RELEASE NOTES FOR MARI 2.0v2

**Release Date** 13 August 2013

## Best Bits of 2.0v2

- Caching and uncaching multiple selected layers now works on all selected layers instead of the last selected layer only.
- Add Mask Stack and Add Adjustment Stack buttons have been added to the Layers palette.
- New functions have been added to the Python API and the API documentation has been updated.

## System Requirements

### Officially Supported Operating Systems

- Windows 7 (64-bit) or above
- Linux 64-bit operating system (Fedora 12 or Red Hat Enterprise Linux 5.4)



**NOTE:** Some newer Linux distributions have changed their network device configuration handling and won't work with Mari without some configuration changes. Known affected distributions at this time include Fedora 16, Red Hat Enterprise Linux 6, and CentOS 6 on certain, notably Dell, hardware. Please see our FAQs for more information: [http://www.thefoundry.co.uk/support/faqs/#question\\_97](http://www.thefoundry.co.uk/support/faqs/#question_97)

### Minimum Hardware Requirements

- Quad-core processor
- 250GB disk space available for caching and temporary files (or a minimum of 50GB if you're working on a small project)
- At least 4GB RAM
- Display with 1680 x 1050 pixel resolution
- An NVIDIA or AMD graphics card with at least 1GB of RAM and OpenGL 3.2 support (see [Tested Graphics Cards](#) below)



**NOTE:** Displacement preview is currently only available on the cards and drivers that support OpenGL 4.0 or newer.

## Virtual Memory Requirements

It is recommended that virtual memory be available for Mari to use, especially when handling large projects. The use of virtual memory improves stability, helping to prevent data loss on large projects.

## Tested Graphics Cards

- NVIDIA GeForce GTX 480
- NVIDIA GeForce GTX 580
- NVIDIA GeForce GTX 680
- NVIDIA Quadro (Fermi Series) 4000
- NVIDIA Quadro (Fermi Series) 5000
- NVIDIA Quadro (Fermi Series) 6000
- NVIDIA Quadro K2000
- NVIDIA Quadro K4000
- NVIDIA Quadro K5000
- AMD FirePro W7000
- AMD FirePro W8000
- AMD FirePro W9000
- AMD Radeon HD 7850
- AMD Radeon HD 7950

Please download and install the latest graphics driver for your card from the NVIDIA or AMD websites.



**NOTE:** Ensure that you are using 8.982.8.1 (AMD Catalyst 12.10) drivers or higher for AMD cards.

Due to the number of cards available in the market, we are unable to test every card and, therefore, cannot officially qualify all of them individually. However, if your machine/GPU specifications meet the requirements mentioned, theoretically Mari should work fine for you. If you encounter any issues, please contact [support@thefoundry.co.uk](mailto:support@thefoundry.co.uk).

## New Features

There are no new features in this release.

## Feature Enhancements

- BUG ID 33877 - The **Layers** palette has been extended and now includes buttons to **Add Mask Stack** and **Add Adjustment Stack**. This makes it easier than navigating through several menus.
- BUG ID 34818 - **Share As Channel** has been renamed to **Share Layers As Channel**, which now shares all selected layers into a new channel.

## Bug Fixes

- BUG ID 34619 - On some operating systems, when creating a new Ptex project, Mari lost its ability to use OpenGL.
- BUG ID 34674 - Windows: Channel Presets did not generate the correct templates when scanning directories.
- BUG ID 34700 - Caching and uncaching multiple selected layers only cached or uncached the last selected layer instead of all those selected. **Cache Layer** and **Uncache Layer** have been renamed to **Cache Layers** and **Uncache Layers** respectively. They now operate on all selected layers instead of the current layer only.
- BUG ID 34885 - The **Run Script** action in the right-click menu in the **Image Manager** was missing.
- BUG ID 35050 - Pressing the corresponding modifier keys while the mouse button or tablet pen were held down did not engage navigation.
- BUG ID 35082 - When shader creation failed it sometimes caused Mari to crash.
- BUG ID 35094 - Mari sometimes crashed after flattening a layer.
- BUG ID 35218 - Channel Presets always resized the images to fit the patches, regardless of the setting in the dialog.
- BUG ID 35237 - **Import into New Channel** did not correctly set the default resolution for the new channel when the **Resize patches to fit images** option was selected.
- BUG ID 35242 - **Share Layers as Channel** did not correctly set the default image resolution for the new channel.
- BUG ID 35728 - Undo stopped working after using the **Paint Buffer Transform** tool.
- BUG ID 35745 - Flattening or merging layers sometimes caused Mari to crash.
- BUG ID 36033 - Patches were being rendered with no color bleed when layers were flattened or cached.

- BUG ID 36393 - Some combinations of **Bake Render Size**, **Virtual Texture Size**, and assets, when cached or flattened, produced incorrect blocks of color.
- BUG ID 36399 - A progress dialog appeared during Python export function calls.
- BUG ID 36484 - In certain instances, baking was corrupted when navigating when **Auto Bake and Clear** was active. Made the canvas widget track the modifier keys used in triggering the various navigation tools, and made it call bake when the corresponding keys are held down, and the auto bake and clear setting is enabled.
- BUG ID 36701 - Converting a layer to paintable with different sized patches caused Mari to become unresponsive.
- BUG ID 37288 - When using the tow brush, shader baking, such as layer caching, flattening, and grouping, resulted in patches not bleeding correctly to expose UV seams.

## Known Issues & Workarounds

### Mari Tools

- BUG ID 15810 - The black borders at the edge of the canvas in **Perspective** view are selected, if an object overlaps the borders when a selection is made using the **Marquee Select** tool, with the selection mode set to **Magic Wand**.
- BUG ID 13640 - The **Blur** tool can be slow to use on the initial stroke. Wait for Mari to process the blur before applying a second stroke.
- BUG ID 13481 - Making a selection using the **Marquee Select Tool** with the selection mode set to **Magic Wand** sometimes ignores regions that are completely surrounded by other selected regions, despite differences in color.
- BUG ID 13394 - Using the **Select Items** tool with the **Facing** set to **Front** to select and hide a portion of faces causes some of the faces within the selection to remain visible when zoomed in.  
To catch all selected faces, either:
  - select **Facing > Through** instead of **Front**,

**OR**

  - zoom in closer to the object.

### Shaders

- BUG ID 34729 - Mari displays a rendering error on the canvas when it is unable to create a shader. More information has been included to help you determine the cause of the error. Some solutions might be to hide

groups and layers, or to cache parts of your layer stack until a shader can be created.

- BUG ID 34679 - On extremely large projects, issues can arise with shader limits and reaching the maximum allowed texture slots available. To avoid reaching these shader limits on large projects, try the following workarounds:
  - hide groups and layers, or
  - cache groups and layers.

### Layers

- BUG ID 34690 - Flattening or caching layers or channels on complex projects may cause Windows to reset the graphics driver due to the long processing time. To work around this issue, you can try to flatten or cache fewer layers at a time, or reduce the value of the **Max Render Size For Baking** setting. This setting can be found under **Preferences > GPU > Baking and Projection**.

Reducing this size breaks the flattening or caching operation up into smaller pieces, which individually take less time to calculate, and thereby avoids a Windows graphics driver reset.

- BUG ID 26460 - Painting a mask in a Mask Layer Group sometimes results in unexpected paint results. To prevent this from happening, either:
  - Use a white "color" layer at the bottom of your mask stack. Any layer used over this initial "color" layer should then be fine,

**OR**

  - If you want to create a mask in a Mask Layer Group, simply add another layer on your Mask Layer Group instead, and paint white into it to create a mask.

### Importing and Exporting

- BUG ID 29386 - When using the **Export for Maya** script, Maya's viewport may incorrectly show some patches as transparent. This can be resolved by selecting **High Quality Rendering** or **Viewport 2.0** from the Renderer menu within Maya.
- BUG ID 16324 - Windows only: you cannot currently import an image into a channel using a relative file path. To work around this, use an absolute path when importing images.
- BUG ID 14985 - There may be a slight pause after importing textures when creating new projects while Mari saves the project.

### Nuke<>Mari Bridge

- BUG ID 23010 - Nuke<>Mari Bridge: If Mari crashes when receiving incoming components from Nuke when the **Virtual Texture Type** is set to **Float**, lower the **Virtual Texture Size** to a value below 8192x8192.
- BUG ID 19829 - Nuke<>Mari Bridge: Unprojecting a displaced piece of geometry does not project correctly in Nuke.
- BUG ID 19780 - Nuke<>Mari Bridge: A projector created in **Ortho** view in Mari does not re-project correctly in Nuke.

### Python

- BUG ID 16616 - Python: PyQt sometimes crashes when adding temporary objects to layouts, or compound widgets such as QWidget. This is because of incorrect reference counting. To work around the problem, always store a reference to every GUI item until you are sure it is no longer needed.
- BUG ID 15491 - Python: PyQt bindings are missing for some enumerations and types such as QFileSystemModel. For enumerations, you can usually work around the problem by passing in the integer value instead.

### Ptex

- BUG ID 17626 - It can take a long time to import very large or very high polygon count Ptex models.  
The workaround is to assign a small uniform face size (1x1 or 2x2) on import, and then increase the resolution of the relevant bits of the model as necessary after loading.
- BUG ID 17618 - Ptex does not bake properly if the resolution of the face is too small.  
The workaround is to increase the resolution of the selected faces you are having problems with.

### Graphics Cards

- BUG ID 34804 - An infrequent issue on Windows AMD graphics card drivers causes shader baking actions, such as flattening, merging, and caching, to swap around parts of the texture. We're working closely with AMD to resolve the problem.  
If you do find this occurring, increasing the **Max Render Size for Baking** setting under **Preferences > GPU > Baking and Projection** may help reduce the frequency of this problem.

- BUG ID 18457 - Using NVIDIA graphics cards from the **Fermi** series with drivers older than version 270 results in various rendering issues when the **Virtual Texture Type** is set to **Half** or **Float**.  
To resolve this, please download and install the latest graphics driver for your card from the NVIDIA website.
- BUG ID 12567 - Enabling **Sync to VBlank** in NVIDIA settings can drastically reduce Mari's performance. If you experience very slow interaction, even with low-polygon models, on one of the [Tested Graphics Cards](#), navigate to:
  - Linux: **NVIDIA X Server Settings > X Screen 0 > OpenGL Settings** and turn off **Sync to VBlank**.
  - Windows: **NVIDIA Control Panel > 3D Settings > Manage 3DSettings > Vertical Sync > Force off**Then, restart Mari.

### Miscellaneous

- BUG ID 33293 - Linux: Launching Mari with the language set to one without certain character symbols resulted in Mari failing with an error that the specified transform could not be loaded.  
To work around this, set the locale (language) to English.
- BUG ID 31946 - Sometimes paint is not baked because of memory management issues on the graphics card. This can be due to issues such as a high resolution paint buffer, a high bit-depth paint buffer, large virtual texture size, or even a large scale value on the paint buffer transform. These issues can usually be identified by **glError: 0x505 out of memory** messages in the log.  
Try reducing any or all of these values to prevent it happening. Graphics drivers are continually improving, so it's also worth checking whether upgrading your drivers resolves the problem.
- BUG ID 20510 - If you find that the startup time for Mari is longer than usual, please check that the LIC files in your RLM licensing data folder do not refer to obsolete server ports. If they do, place them in another directory and restart Mari.
- BUG ID 20021 - Textures in the canvas intermittently switch between lower and higher resolutions.  
This issue is more likely to occur if your virtual texture resolution is low, and you're working on a complex model with displacement. Possible work-arounds include increasing your virtual texture size, reducing the number of channels Mari has to access at once (for example, by reducing the number of channels required for the current shader), to reduce the patch resolution of patches in the channels used in the shader, or to use a smaller canvas window or monitor.

- BUG ID 14201 - Linux only: Mari becomes unresponsive after the system is woken from sleep.
- BUG ID 13700 - Adjusting the **Camera > Perspective** settings for a **Projector** is not reflected on the canvas until the **Projector** is made **Current**.
- BUG ID 13571 - Launching a new version of Mari for the first time, when a config file exists from a previous version, sometimes results in an object not appearing in the Ortho view.

To solve this, close Mari, delete the following config file and relaunch Mari:

- Linux: `~/.config/TheFoundry/Mari.conf`
- Windows: `C:/Users/<login>/.mari/TheFoundry/Mari.ini`
- BUG ID 13294 - Windows: Mari sometimes crashes when trying to load data on large projects due to the program exhausting all Window manager objects.

To reconfigure the user object limit:

- Open regedit and navigate to **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\NT\CurrentVersion\Windows**, and
- Edit **USERProcessHandleQuota** to a larger number.

If this number gets too large, you may also have to modify **GDIPProcessHandleQuota**.

- BUG ID 12102 - Current brush settings do not get saved as part of the project. Instead, Mari reverts to the default settings when you close and relaunch it.
- BUG ID 11874 - Mari doesn't recognize 3-digit padded **.obj** sequences as animation.



## Developer Notes

These are the changes relevant to developers.

### New Features

- BUG ID 35012 - The `mari.Channel.setName()` and `name()` functions have been added to the Python API.
- BUG ID 35941 - The `Channel` and `LayerStack` classes are now derived from the `Metadata` class, so that metadata can be added to a channel, as in earlier versions.
- BUG ID 36142 - The `Layer.siblingSharedLayerList()` function has been added. This function returns the list of other layers that share resources with this layer.
- BUG ID 36165 - The following functions have been added to the Python API:
  - `AdjustmentLayer.primaryAdjustmentType()`
  - `AdjustmentLayer.secondaryAdjustmentType()`
  - `ProceduralLayer.proceduralType()`.

These provide a reliable way to retrieve the types of adjustment or procedural layers, and in a form that is more consistent with other functions. The existing similar functions, such as `proceduralName()`, have now been deprecated.

### Feature Enhancements

- BUG ID 34698 - The `Layer.setName()` function has been added to the Python API.
- BUG ID 34727 - The `LayerStack.exportImages()`, `LayerStack.exportImagesFlattened()`, `PaintableLayer.exportImages()`, and `ProceduralLayer.exportImages()` functions all accept an optional list of UV indices to export.
- BUG ID 34805 - The `Added Layer.nameChanged`, `Channel.nameChanged`, and `Shader.nameChanged` signals have been added to the Python API.
- BUG ID 35031 - The `Shader.setName()` and `name()` functions have been added to the Python API.
- BUG ID 35142 - The Indices page of the Python API documentation now appears on a single page, as it was in pre-2.0v1 versions.
- BUG ID 36169 - `LayerStack.adjustmentLayerTypeList()` and `LayerStack.proceduralLayerTypeList()` are now static, so they can be called through `mari.LayerStack` instead of requiring a `LayerStack` object.
- BUG ID 36250 - Metadata can be added to shader classes in the Python API.

- BUG ID 36252 - The **GeoEntity.shaderStandaloneTypeList()**, **GeoEntity.shaderDiffuseTypeList()**, and **GeoEntity.shaderSpecularTypeList()** functions have been made static. This allows them to be called without a specific GeoEntity object.

## Bug Fixes

- BUG ID 15922/36208 - The **ImageSet.importImages()** function did not properly handle \$CHANNEL and \$LAYER token replacements in file names during imports.  
The **ImageSet.importImage()** function can now be called with a file name template and an optional list of UV indices to import images for.
- BUG ID 35294 - Adjustment layer vector parameters, such as in **Grade** adjustments, were not properly processed through Python. These can now be set from lists or **mari.VectorN** objects, and are returned as lists.
- BUG ID 35940 - The **Channel.flatten()** function did not return the resulting layer.